CORE-PERIPHERY PRINCIPLE GUIDED REDESIGN OF SELF-ATTENTION IN TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Designing more efficient, reliable, and explainable neural network architectures is a crucial topic in the artificial intelligence (AI) field. Numerous efforts have been devoted to exploring the best structures, or structural signatures, of well-performing artificial neural networks (ANN). Previous studies, by post-hoc analysis, have found that the best-performing ANNs surprisingly resemble biological neural networks (BNN), which indicates that ANNs and BNNs may share some common principles to achieve optimal performance in either machine learning tasks or cognitive/behavior processes. Inspired by this phenomenon, rather than relying on post-hoc schemes, we proactively instill organizational principles of BNNs to guide the redesign of ANNs by infusing an efficient information communication mechanism of BNNs into ANNs. Specifically, we quantified the typical Core-Periphery (CP) organization of the human brain networks, infused the Core-Periphery principle into the redesign of vision transformer (ViT), and proposed a novel CP-ViT architecture: the pair-wised densely interconnected self-attention architecture of ViT was upgraded by a sparse Core-Periphery architecture. In CP-ViT, the attention operation between nodes (image patches) is defined by a sparse graph with a Core-Periphery structure (CP graph), where the core nodes are redesigned and reorganized to play an integrative role and serve as a center for other periphery nodes to exchange information. We evaluated the proposed CP-ViT on multiple public datasets, including medical image datasets (INbreast) and natural image datasets (CIFAR-100). We show that there exist sweet spots of CP graphs that lead to CP-ViTs with significantly improved performance. In general, our work advances the state of the art in three aspects: 1) This work provides novel insights for brain-inspired AI: we can instill the efficient information communication mechanism of BNNs into ANNs by infusing similar organizational principles of BNNs into ANNs; 2) The optimized CP-ViT can significantly improve its predictive performance while dramatically reducing computational cost by benefiting from the infused efficient information communication mechanism existing in BNNs; and 3) The core nodes in CP-ViT can identify task-related meaningful and important image patches, which can significantly enhance the interpretability of the trained deep model. (Code is ready for release).

1 INTRODUCTION

Aided by the rapid advancement in hardware and massively available data, deep learning models have witnessed an explosion of various ANN architectures He et al. (2016); Krizhevsky et al. (2017); Vaswani et al. (2017), and made breakthroughs in many application fields due to their powerful automatic feature extraction capabilities. It has been proven that the design of ANN architecture, particularly the neuron wiring patterns and their message exchange mechanisms, can play a critical role in the feature representation and the downstream task performance, e.g., in convolutional neural networks (CNNs)Simonyan & Zisserman (2014); Szegedy et al. (2015); Ren et al. (2015); Krizhevsky et al. (2017) and recurrent neural networks (RNNs) Gers et al. (2000); Cho et al. (2014). More recently, transformer has become another mainstream ANN architecture due to its outstanding self-attention mechanism that allows effective and efficient message exchanges among neurons, and produced promising results in the natural language processing Vaswani et al. (2017); Devlin et al. (2018) and computer vision domains Dosovitskiy et al. (2020); Liu et al. (2021). In particular, many advancements in transformer architecture design, e.g., vision transformer (ViT) Dosovitskiy et al. (2020), have centered around more effective message exchange mechanisms among spatial tokens by designing different Token Mixers. For instance, the shifted window attention in Swin Liu et al. (2021), the token-mixing MLP in Mixer Tolstikhin et al. (2021), and the pooling in MetaFormer Yu et al. (2022), among others, were all designed to improve the self-attention upon the original vanilla ViT Dosovitskiy et al. (2020), and thus enable more effective and efficient message exchanges among spatial patches/tokens. However, despite tremendous advancements in ANN architecture design in CNNs, RNNs, and transformers, particularly for better message exchange mechanisms, there has been a fundamental lack of general principles that can inform and guide such ANN architecture design and redesign.

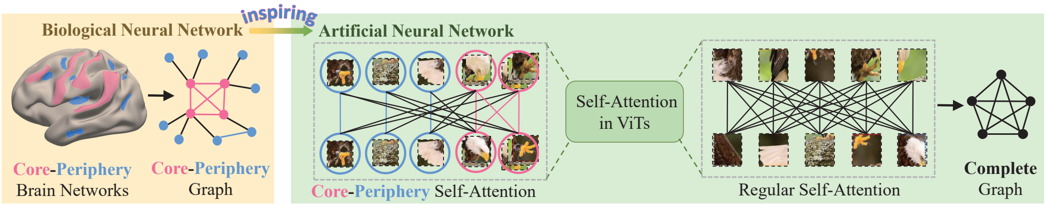


Figure 1: The Core-Periphery principle in brain networks inspires the design of ANNs. The Core-Periphery structure broadly exists in brain networks, with a dense “core” of nodes (pink) densely interconnected with each other and a sparse “periphery” of nodes (blue) sparsely connected to the core and among each other. Inspired by this principle of BNN, we aim to instill the Core-Periphery structure into the self-attention mechanism and propose a new CP-ViT model.

To seek such guiding principles for ANN architecture design, more and more research studies started with exploring the “structural signatures” of well-performing ANNs. The deep learning community has witnessed a paradigm shift from optimal feature design to optimal ANN architecture design. In general, the major strategies for optimal ANN architecture design can be categorized into two basic streams based on how to search in the neural architecture space. The first strategy is to design neural architectures that achieve the best possible performance using given computing resources in an automated way with minimal human intervention. Neural architecture search (NAS) Zoph & Le (2016); Ren et al. (2021); Elsken et al. (2019) is a mainstream methodology in this category. The advantage of the first category is obvious: it has relatively low demand on the researchers’ prior knowledge and experience, making it easier to perform modifications to the neural architecture in line with their actual tasks. However, these methods usually come with a high computational cost. The second category of strategy is to take the advantage of prior knowledge from specific domains, such as brain science, to guide ANN architecture design. For example, the authors in Zhang et al. (2021) designed a two-stream model for grounding language learning in vision based on the brain science principle that humans learn language by grounding concepts in perception and action, and encoding “grounded semantics” for cognition. It is worth noting that the above-mentioned two strategies should be viewed as complementary to each other rather than being in conflict, and their combination provides the researchers with an opportunity to explore and design well-performing neural architectures under different principles. For instance, recent studies, via qualitatively post-hoc analysis, have found that the best-performing ANNs surprisingly resemble BNNs You et al. (2020), which indicates that ANNs and BNNs may share some common principles to achieve optimal performance in either machine learning tasks or cognition/behavior processes.

Inspired by the above-mentioned prior outstanding studies, in this work, we aim to proactively instill the organizational principle of Core-Periphery structure in BNNs to guide the redesign of ANNs by using ViT as a working example. The concept of Core-Periphery brain network is illustrated in Figure 1. It has been widely confirmed that the Core-Periphery pattern universally exists in the functional networks of human brains and other mammals, effectively promoting the efficiency of information transmission and communication for integrative processing Bassett et al. (2013); Gu et al. (2020). By using the Core-Periphery property as a guiding principle, we infused its effective and efficient information communication mechanism into the redesign of ViT. To this end, we quantified the Core-Periphery property of the human brain network, infused the Core-Periphery property into ViT, and proposed a novel CP-ViT architecture. Specifically, we upgrade the complete graph of dense connections in the original vanilla ViT Dosovitskiy et al. (2020) with a sparse graph with

Core-Periphery property (CP graph), where the core nodes are redesigned and reorganised to play an integrative role and serve as a center for other periphery nodes to exchange information. Moreover, in our design, a novel learning mechanism is used to endow the core nodes with the power to identify and capture the task-related meaningful and important image patches. We evaluated the proposed CP-ViT on multiple public datasets, including medical image datasets (INbreast) and natural image datasets (CIFAR-100). The results suggest that the optimized CP-ViT in sweet spots You et al. (2020) outperforms other ViTs. We summarize our contributions in three aspects: 1) This work provides novel insights for brain-inspired AI: we can instill the efficient information communication mechanism in BNNs into ANNs by infusing similar organizational principles of BNNs into ANNs; 2) The optimized CP-ViT can significantly improve its predictive performance while dramatically reducing computational cost, benefiting from the infused efficient information communication mechanism; and 3) The core nodes in CP-ViT can identify task-related meaningful and important image patches, which can significantly enhance the interpretability of the trained deep model.

2 Related Work

Core-periphery Structure The Core-Periphery structure is a fundamental network signature that is composed of two qualitatively distinct components: a dense “core” of nodes strongly interconnected with one another, allowing for integrative information processing to facilitate the rapid transmission of message, and a sparse “periphery” of nodes sparsely connected to the core and among each other Gallagher et al. (2021). The Core-Periphery pattern has helped explain a broad range of phenomena in network-related domains, including online amplification Barberá et al. (2015), cognitive learning processes Bassett et al. (2013), technological infrastructure organization Alvarez-Hamelin et al. (2005); Carmi et al. (2007), and critical disease-spreading conduits Kitsak et al. (2010). All these phenomena suggest that the Core-Periphery pattern may play a critical role to ensure the effectiveness and efficiency of information exchange within the network. In the literature, there are two widely-used approaches for generating graphs with Core-Periphery property (CP graphs): the classic two-block model of Borgatti and Everett (BE algorithm) Borgatti & Everett (2000) and the k-cores decomposition Gallagher et al. (2021). The former approach partitions a network into a binary hub-and-spoke layout, while the latter one divides it into a layered hierarchy. In this work, for simplicity, we adopted a two-block model to generate a CP graph which is used to guide the self-attention operations between patches (tokens) in ViT. In this way, the Core-Periphery property is infused into the ViT model.

Methods for Designing More Efficient ViT Architecture ViT and its variants have achieved promising performances in various computer vision tasks, but their gigantic parameter-counts, heavy run-time memory usage, and high computational cost become a major burden for the application. Therefore, there exists an impending need to develop lightweight and efficient vision transformers. For this purpose, several studies aimed to use network pruning, sparse training, and supernet-based NAS to slim vanilla ViT. **From token level**, Tang et al. Tang et al. (2022) designed a patch slimming method to discard useless tokens. Evo-ViT Xu et al. (2022) updated the selected informative and uninformative tokens with different computation paths. VTP Zhu et al. (2021) reduced embedding dimensionality by introducing control coefficients. **From model architecture level**, UP-ViTs Yu & Wu (2021) pruned the channels in ViTs in a unified manner, including residual connections in all the blocks, MHSAs, FFNs, normalization layers, and convolution layers in ViT variants. SViTE Chen et al. (2021b) dynamically extracted and trained sparse subnetworks instead of training the entire model. To further co-explore data and architecture sparsity, a learnable token selector was used to determine the most vital image patch embeddings in the current input sample. AutoFormer Chen et al. (2021a) and ViTAS Su et al. (2021) leveraged supernet-based NAS to optimize the ViT architecture. Despite the remarkable improvements achieved by the above methods, both token-sampling and data-driven strategies may highly depend on the data and tasks performed, impeding the vision transformers’ generalization capability. A more universal principle (e.g., derived from BNNs) that can guide a more efficient design of ANN’s architecture is much desired. In this work, we will leverage a widely existing Core-Periphery property in BNN to develop a more efficient CP-ViT.

3 Core-Periphery Principle Guided Transformers

The Core-Periphery principle can be applied to ViT and its variants via a unified framework that is illustrated in Figure 2. The framework has two main parts, Core-Periphery graph generation and Core-Periphery graph guided re-design of self-attention mechanism.

3.1 Core-Periphery Graph Generation

The self-attention of our proposed CP-ViT is controlled by Core-Periphery graphs (CP graphs). We proposed a Core-Periphery graph generator to generate a large spectrum of CP graphs in the graph space defined by the total node number and the core node number. Although several graph generators have been proposed in previous works, they were not designed for generating CP graphs. For example, Erdos-Renyi (ER) generator samples graphs with given node and edge numbers uniformly and randomly Erdos et al. (1960); Watts-Strogatz (WS) generator generates graphs with small-world properties Watts & Strogatz (1998); and complete graphs generator generates graphs where nodes are pair-wise densely connected with each other Skiena (1991).

To generate graphs with the Core-Periphery property, we proposed a novel CP graph generator which is parameterized by a total node number n, a core node number m, and three wiring thresholds p_{cc}, p_{cp}, p_{pp} which are the wiring probabilities between core-core nodes, core-periphery nodes, and periphery-periphery nodes, respectively. Based on these measures, the CP graph generation process is as follows: we first defined the core nodes number m and the periphery nodes number $n-m$; Then, for core-core node connections, we traversed all core-core node pairs. And for each of them, we used a random seed sampled from the continuous uniform distribution in $[0, 1]$ to generate a wiring probability p_{rs}, and if the wiring probability is greater than the threshold p_{cc}, the two core nodes are connected. This wiring process is formulated as:

$$A\left(i, j\right) = \begin{cases} 1 & \text{if } p_{rs} \geq p_{cc} \\ 0 & \text{if } p_{rs} < p_{cc} \end{cases} \tag{1}$$

where A is the adjacency matrix of the generated graph, 1 means that there exists an edge between nodes i and j, 0 means there is no edge. The same procedure was applied to core-periphery and periphery-periphery node pairs with the corresponding thresholds p_{cp} and p_{pp}, respectively. In this way, by using different n and m and wiring thresholds, we can generate different graphs extensively in the graph space; finally, all the generated graphs were examined by the Core-Periphery detection algorithm (BE algorithm) Borgatti & Everett (2000) and the graphs with the Core-Periphery property will be used in further steps to guide the self-attention operation.

3.2 Core-Periphery Guided Self-Attention

To instill the Core-Periphery principle into the self-attention mechanism in ViT, we redesigned the self-attention operations according to the generated CP graphs where the patches are replaced by the nodes, and the new self-attention relations are replaced by the edges in the CP graph. With this representation paradigm, a complete graph can represent the self-attention of the vanilla ViT, and similarly, the Core-Periphery principle can be effectively and conveniently infused into the ViT architecture by upgrading the complete graph with the generated CP graphs. The new self-attention rules can then be redefined: CP graph can be represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with nodes set $\mathcal{V} = \{\nu_1, ..., \nu_n\}$, edges set $\mathcal{E} \subseteq \{(\nu_i, \nu_j)|\nu_i, \nu_j \in \mathcal{V}\}$, and adjacency matrix A. The CP graph guided self-attention for a specific node i at r-th layer of CP-ViT is defined as:

$$x_i^{(r+1)} = \sigma^{(r)}(\{(\frac{q_i^{(r)}(K_j^{(r)})^T}{\sqrt{d_k}})V_j^{(r)}, \forall j \in N(i)\}) \tag{2}$$

where $\sigma(\cdot)$ is the activation function, which is usually the softmax function in ViTs, $q_i^{(r)}$ is the query of patches in the i-th node in $\mathcal{G}$, $N(i) = \{i|i \vee (i,j) \in \mathcal{E}\}$ are the neighborhood nodes of node i, d_k is the dimension of queries and keys, and $K_j^{(r)}$ and $V_j^{(r)}$ are the key and value of patches in node j.

In CP-ViT, each node can contain a single patch or a set of multiple patches. We proposed the following patch assigning pipeline to map the original patches to the nodes of the CP graph. In vanilla ViT, one input image is divided into $16 \times 16 = 196$ patches. When we use a CP graph with

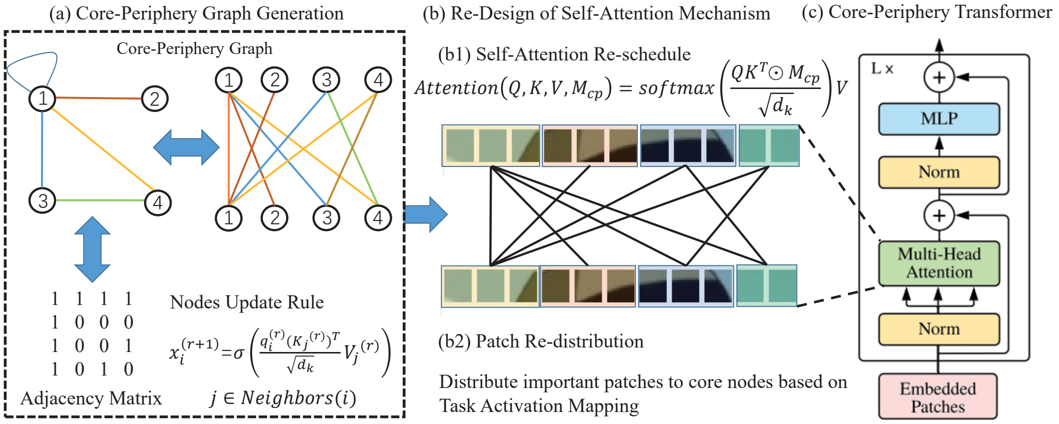

(a) Core-Periphery Graph Generation
(b) Re-Design of Self-Attention Mechanism
(c) Core-Periphery Transformer
Core-Periphery Graph
1
2
3
4
(b1) Self-Attention Re-schedule
$Attention(Q,K,V,M_{cp}) = softmax\left(\frac{QK^T \odot M_{cp}}{\sqrt{d_k}}\right)V$
L ×
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches
1 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
Adjacency Matrix
Nodes Update Rule
$x_i^{(r+1)} = \sigma\left(\frac{q_i^{(r)}(K_j^{(r)})^T}{\sqrt{d_k}} V_j^{(r)}\right)$
$j \in Neighbors(i)$
(b2) Patch Re-distribution
Distribute important patches to core nodes based on Task Activation Mapping

Figure 2: Core-Periphery Principle Guided Re-design of Self-Attention. The proposed Core-Periphery guided re-design framework for ViTs consists of two major components: the Core-Periphery graph generator and re-design of the self-attention mechanism. The basic idea is that we mapped the ViT structure to graphs and proposed a new graph representation paradigm to represent the self-attention mechanism. Under this paradigm, the design of self-attention mechanism can be turned into a task of designing desirable graphs. (a) The CP graph generator was proposed to generate graphs with Core-Periphery property in a wide range of search spaces. (b) The self-attention of the nodes is controlled by the generated CP graph and the patches are re-distributed to different nodes by a novel patch distribution method. (c) The new self-attention mechanism will upgrade the regular self-attention in vanilla ViT. The new ViT architecture is thus named as CP-ViT

n nodes to design the self-attention mechanism, 196 mod n nodes will be assigned $\lfloor 196/n \rfloor + 1$ patches and the remaining $n - (196 \; mod \; n)$ nodes will be assigned $\lfloor 196/n \rfloor$ patches. For example, if we use a 5 node CP graph, the 5 nodes will have 40, 39, 39, 39, and 39 patches, respectively; and if we use a 196 nodes CP graph in another case, each node will contain a single patch. The patches are randomly assigned to the nodes at the beginning of the training process and then they will be re-distributed iteratively after each training epoch based on a novel patch distribution method to be elaborated in the next section. Based on the above discussion, the CP graph guided self-attention that is conducted at the node level can be formulated as:

$$Attention(Q, K, V, M_{cp}) = softmax(\frac{QK^T \odot M_{cp}}{\sqrt{d_k}}V) \tag{3}$$

where queries, keys, and values of all patches are packed into matrices Q, K, and V, respectively, M_{cp} is the mask matrix derived from the adjacency matrix A of the CP graph, and $\odot$ is the dot product. The derivation of M_{cp} from A is detailed in the appendix.

3.3 Patch Redistribution

The organization of Core-Periphery structure will make the information communication and message exchange at core nodes more intensive while less frequently at periphery nodes. This is based on the foundation that the core nodes usually process the basic information in many biological networks Bassett et al. (2013). To this end, we need to evaluate the importance of the patches and select the most important ones to assign to the core nodes, which is defined as task activation mapping.

For a specific task of CP-ViT, in order to identify the important patches, we computed the gradients of the output y (before the activation function) with respect to patch features (after patch embedding) P^k, i.e. $\frac{\partial y}{\partial P^k}$. These gradients flowing back to the patch features are global-average-pooling over the feature dimensions to obtain the patch importance weights. The important weights are:

$$\alpha_k = \frac{1}{Z} \sum_{i=1}^{Z} \frac{\partial y}{\partial P_i^k} \tag{4}$$

where Z is the dimension of the patch embedding features. Then top K patches that have the highest importance weights are selected and re-distributed to the core nodes. Note that the patch distribution process is not random but distributed based on the nodes' degree in a descending manner: the patches with higher importance weights are distributed to the core nodes with a higher degree.

4 Experiments

4.1 Exploring Core-Periphery Graphs

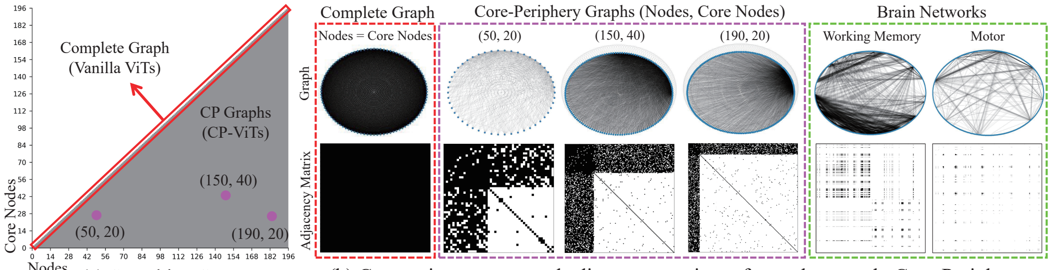
Core-Periphery property in brain networks. We quantitatively measured the Core-Periphery property in two typical functional brain networks under working memory (BN-WM) and motor (BN-M) tasks. Specifically, the functional brain networks under specific task were created by using task fMRI data. Each fMRI voxel represents a tidy cube of brain tissue — a 3D image building block. Using voxels as nodes and the correlations between two fMRI signals of each pair of voxels as edges, we generated two population level functional networks and showed their connection patterns and adjacency matrix in Figure 3(b). To measure the Core-Periphery property of the two functional brain networks, we adopted independent probability Cucuringu et al. (2016). Independent probability is defined as the probability that there is an edge between any pairs of nodes in a given matrix. It is an important measurement to indicate if the matrix or graph is organized in Core-Periphery manner Holme (2005) Rombach et al. (2014). According to previous studies Liu et al. (2019), the convex gyri and the concave sulci areas of the cerebral cortex function as the "core" and "periphery", respectively, in the process of the brain's functional networking. Taking voxels in gyri areas as core and in sulci areas as periphery, we calculated the independent probabilities I_{cc}, I_{pp} and I_{cp} for core-core connections, periphery-periphery connections, and core-periphery connections, respectively, and showed the results in Table 1. We can see that $I_{cc} > I_{cp} > I_{pp}$, which suggests that the proposed CP graphs have true Core-Periphery structure.

Core-Periphery structure in artificial neural networks. We introduced the Core-Periphery organization into ANNs by CP graphs. As discussed in Sections 3.1 and 3.2, there are two key factors that have an important influence on the CP graph generation process. One is the node number, including the total nodes number and core nodes number, and the other is the wiring thresholds. The total nodes number and the core nodes number define the whole search space that contains $\sum_{i=1}^{196}\sum_{j}^{0<j<=i}(i+j) = 19208$ types of CP graphs. The wiring thresholds p_{cc}, p_{cp}, and p_{pp} determine the wiring pattern of a specific CP graph. The search space of CP graphs was shown in Figure 3(a) where the complete graphs located at the diagonal were highlighted by a red box and three types of CP graphs were highlighted by pink circles. The wiring patterns and adjacency matrices of the complete graph and the three CP graphs were shown in Figure 3(b). As shown in Figure 3(b), CP graphs are densely connected in core nodes and sparsely connected in periphery nodes. The overall connection patterns of CP graphs are more sparse than the complete graph. For each type of CP graph, we generated 5 samples with different wiring patterns and obtained 19208 * 5 CP graphs in total. Since the number of the generated CP graphs could be too big (19208 * 5 in total) to explore, we sampled 190 types of CP graphs out of the total 19208 and finally obtained 190*5 candidates. In our experiments, we only adopted the 190*5 candidates to design the CP-ViTs.

Table 1: Evaluation of the Core-Periphery property in CP graphs, graphs generated by other graph generators, and brain networks

IP	CP Graphs	Comp. Graphs	WS Graphs	ER Graphs	BN-M	BN-WM
I_{cc}	0.89 ± 0.06	1.00 ± 0.00	0.48 ± 0.27	0.42 ± 0.23	0.70 ± 0.11	0.72 ± 0.09
I_{cp}	0.53 ± 0.13	1.00 ± 0.00	0.49 ± 0.28	0.42 ± 0.24	0.22 ± 0.07	0.30 ± 0.10
I_{pp}	0.10 ± 0.06	1.00 ± 0.00	0.24 ± 0.28	0.32 ± 0.22	0.07 ± 0.05	0.04 ± 0.04

Similar to brain networks, we also used the independent probability to measure the Core-Periphery property for generated CP graphs. We calculated the averaged independent probability over 190*5 CP graphs and showed the results in Table 1. From the table we can see that $I_{cc} > I_{cp} > I_{pp}$, which suggests that the generated CP graphs also have Core-Periphery property. We further compared our proposed CP graph generator with the following classic graph generators: (1) Erdos-Renyi (ER) generator; (2) Watts-Strogatz (WS) generator; and (3) Complete graph generator. As shown in Table 1, the graphs generated by the classic graph generators don't have Core-Periphery property.



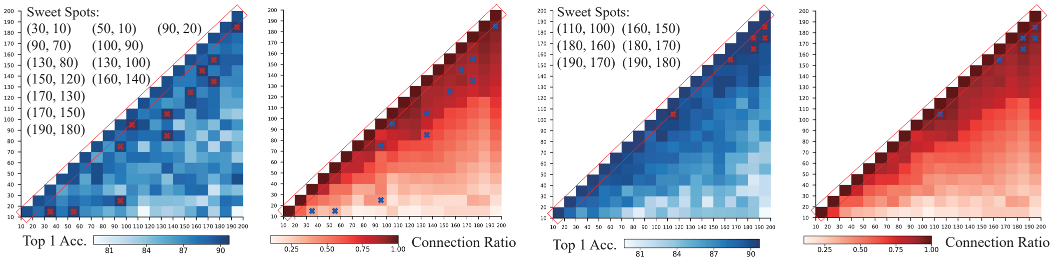
(a) Searching Space

(b) Connection patterns and adjacency matrices of complete graph, Core-Periphery graphs, and brain networks

Figure 3: (a) Graph search space defined by the total nodes number and the core nodes number. The complete graphs are located at the diagonal highlighted by red box and the CP graphs are located at the remaining parts. Three examples of the CP graphs are shown in (b) where the first shows their wiring patterns, and the second row shows their corresponding adjacency matrices. Black color in adjacency matrices means connections between nodes, while white represents no edge. Two types of representative brain networks in motor and working memory tasks are presented in (b).

4.2 Sweet Spots for CP-ViTs

In this section, we evaluated the performance of the proposed CP-ViT. The CP-ViT was implemented based on the ViT-S/16 architecture Chen et al. (2021c) and evaluated on 2 different types of public datasets, the medical image dataset INbreast (Moreira et al. (2012)) and the natural image dataset CIFAR-100 (Krizhevsky et al. (2009)). The parameters of CP-ViT were initialized and fine-tuned from ViT-S/16 trained on ImageNet (Russakovsky et al. (2015)). The introduction of the datasets and experiment settings are in the appendix.



(a) Acc. of CP-ViT on INbreast (b) CR of CP-ViT on INbreast (c) Acc. of CP-ViT on CIFAR-100 (d) CR of CP-ViT on CIFAR-100

Figure 4: Performance of CP-ViT measured using INbreast and CIFAR-100 datasets. Sub-figures (a) and (c) show the classification accuracy of the CP-ViTs and vanilla ViTs in the search space. A deeper color means higher top-1 accuracy. Sweet spots are marked by red crosses, in which CP-ViTs achieve better performance than vanilla ViT. Sub-figures (b) and (d) are the self-attention connection ratio of the CP-ViTs and vanilla ViT. Lighter color means a lower connection ratio. Sweet spots are marked by the blue crosses.

We explored the performance of different types of CP graphs in the search space (Figure 3(a)) in terms of top 1 accuracy and connection ratio. The connection ratio (CR) quantitatively measures the computational costs of different self-attention operations, which is defined by (5):

$$CR = \frac{|1|_{M_{cp}}}{\|M_{cp}\|_1} \tag{5}$$

where $|1|$ represents the number of 1 in the mask matrix of cp graphs, and $\|\bullet\|_1$ is the number of elements in the mask matrix. In general, CR is the ratio of actual self-attention operations to the potential maximum self-attention operations. Given a graph, the potential maximum self-attention operation is fixed. Less actual self-attention operation means less computational cost and hence it has a smaller CR value.

Table 2: Comparison between the proposed CP-ViT with other ViTs, including ViT-S Dosovitskiy et al. (2020), DeiT-B Touvron et al. (2021), and DeiT-T Touvron et al. (2021)

Dataset	Model	CP Graph	CR (%)	Param.(M)	Top1 Acc. (%)
INbreast	ViT-S	–	100.00	21.7	89.91
	CP-ViT	(30, 10)	32.36	16.8	90.58
	CP-ViT	(50, 10)	**29.20**	**16.6**	90.01
	CP-ViT	(90, 20)	43.82	17.6	90.58
	CP-ViT	(90, 70)	84.50	20.6	90.01
	CP-ViT	(100, 90)	92.80	21.2	**90.69**
	CP-ViT	(130, 80)	31.34	19.4	90.58
	CP-ViT	(130, 100)	82.94	20.5	**90.69**
	CP-ViT	(150, 120)	84.18	20.6	90.01
	CP-ViT	(160, 140)	87.77	20.8	90.58
	CP-ViT	(170, 130)	80.79	20.3	90.58
	CP-ViT	(170, 150)	87.65	20.8	90.12
	CP-ViT	(190, 180)	84.89	21.3	90.69
CIFAR-100	ViT-S	–	100.00	21.7	91.20
	DeiT-B	–	100.00	86.6	90.85
	DeiT-T	–	100.00	5.7	85.01
	CP-ViT	(110, 100)	94.49	21.3	**91.45**
	CP-ViT	(160, 150)	**90.48**	**21.0**	91.26
	CP-ViT	(180, 160)	93.67	21.2	91.36
	CP-ViT	(180, 170)	95.84	21.4	91.23
	CP-ViT	(190, 170)	92.59	21.2	91.24
	CP-ViT	(190, 180)	94.89	21.3	91.22

For each specific combination in the search space, we trained the CP-ViT with 5 different CP graph samples and reported the best result in Figure 4. We highlighted the sweet spots with red crosses in Figure 4(a)(c). In Figure 4(a)(c), deeper color means better performance. The performance of CP-ViTs varies in the search space. This result indicates that different self-attention patterns have great influences on the performances of ViTs. Compared to vanilla ViTs with a fully-connected self-attention pattern, the proposed CP-ViT provides the potential for the model to search for optimal self-attention patterns. The CRs of all the ViTs including vanilla ViTs and CP-ViTs were shown in Figure 4(b)(d). The CRs of sweet spots were marked with a blue cross. The results show that besides the improvement in classification accuracy ($+\mathbf{0.78}\%$ for INbreast, $+\mathbf{0.25}\%$ for CIFAR-100), the proposed CP-ViT also leads to a great reduction in computational cost thanks to less self-attention operations ($-\mathbf{70.80}\%$ connections for INbreast, $-\mathbf{9.52}\%$ connections for CIFAR-100). The model setting, top 1 accuracy, and CRs of different ViTs were reported in Table 2. Compared to other ViTs, the CP-ViT can not only improve classification performance but also reduce the computational cost. On CIFAR-100, we compared the performance of the proposed CP-ViT with other variants of ViTs, and the results show that the proposed CP-ViT outperforms other state-of-the-art methods. Notably, the "sweet spots" exist on both INbreast and CIFAR-100 datasets, which indicates that CP-ViTs can improve the classification performance on different datasets and reduce the computational cost. The results demonstrate the superiority of the CP-ViT.

4.3 Visualization of Important Patches

Another advantage of CP-ViTs is that they can identify task-related meaningful and important image patches and effectively improve the interpretability of the deep learning model by allocating these important patches to core nodes (core patches). To evaluate this, we show the patches that were re-distributed to core nodes when the model was well trained in Figure 5. For INBreast, we randomly selected images of three subjects of each class and displayed the original images, the images overlaid with important patches, and images overlaid with expert's eye gazes in three columns. As shown in the figure, the patches of the core nodes are well co-localized with the locations that were identified as diagnostic biomarkers of the disease in literature publications Ibrokhimov & Kang (2022). We also show the medical physicians' eye gaze maps on these images, given that the eye gaze acquired by eye-tracking equipment is considered as the ground truth for identifying important areas in the image. The important patches identified by CP-ViT highly overlap with the eye gaze maps, demonstrating that CP-ViT can effectively identity important meaningful patches. For CIFAR-100, we also visualized the patches assigned to the core nodes under the black dotted line in Figure 5. It is clear that the objects in the patches of core nodes are semantically related to the class labels.

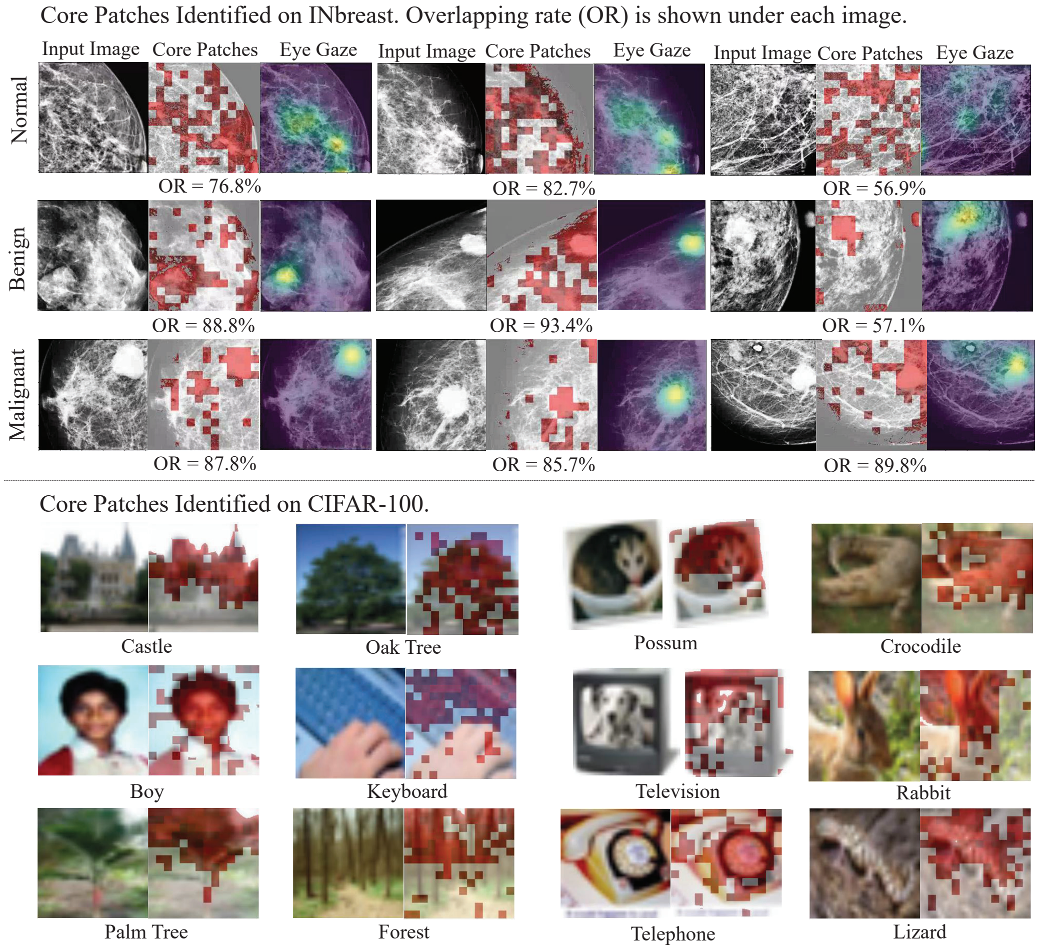


Figure 5: Visualization of important image patches that were distributed to the core nodes. For the INbreast dataset (first block), images of three randomly selected subjects for each class were shown. For each subject, there are three images displayed in three columns. The left column is the original image, the middle column shows the important patches marked by red, and the right column is the eye gaze of medical physicians on the image. For the CIFAR-100 dataset (second block), the important patches identified in the twelve randomly selected classes were displayed.

5 Conclusions

In this work, we proactively instilled an organizational principle of BNN, that is, Core-Periphery property, to guide the design of ANN of ViT. Our extensive experiments suggest that there exist sweet spots of CP graphs that lead to CP-ViTs with significantly improved predictive performance. In general, our work advances the state of the art in three ways: 1) this work provides novel insights for brain-inspired AI by applying organizational principles of BNNs to ANN design; 2) the optimized CP-ViT can significantly improve its predictive performance while dramatically reducing the computational cost; and 3) the core nodes in CP-ViT can identify task-related meaningful image patches, which can significantly enhance the interpretability of the trained deep model.

References

José Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. K-core decomposition of internet graphs: hierarchies, self-similarity and measurement biases. *arXiv preprint cs/0511007*, 2005.

Pablo Barberá, Ning Wang, Richard Bonneau, John T Jost, Jonathan Nagler, Joshua Tucker, and Sandra González-Bailón. The critical periphery in the growth of social protests. *PloS one*, 10 (11):e0143611, 2015.

Danielle S Bassett, Nicholas F Wymbs, M Puck Rombach, Mason A Porter, Peter J Mucha, and Scott T Grafton. Task-based core-periphery organization of human brain dynamics. *PLoS computational biology*, 9(9):e1003171, 2013.

Stephen P Borgatti and Martin G Everett. Models of core/periphery structures. *Social networks*, 21 (4):375–395, 2000.

Shai Carmi, Shlomo Havlin, Scott Kirkpatrick, Yuval Shavitt, and Eran Shir. A model of internet topology using k-shell decomposition. *Proceedings of the National Academy of Sciences*, 104 (27):11150–11154, 2007.

Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12270–12280, 2021a.

Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 34:19974–19988, 2021b.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021c.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Mihai Cucuringu, Puck Rombach, Sang Hoon Lee, and Mason A Porter. Detection of core–periphery structure in networks using spectral methods and geodesic paths. *European Journal of Applied Mathematics*, 27(6):846–887, 2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.

Paul Erdos, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

Ryan J Gallagher, Jean-Gabriel Young, and Brooke Foucault Welles. A clarified typology of core-periphery structure in networks. *Science advances*, 7(12):eabc9800, 2021.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

Shi Gu, Cedric Huchuan Xia, Rastko Ciric, Tyler M Moore, Ruben C Gur, Raquel E Gur, Theodore D Satterthwaite, and Danielle S Bassett. Unifying the notions of modularity and core–periphery structure in functional brain networks during youth. *Cerebral Cortex*, 30(3):1087–1102, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. cvpr. 2016. *arXiv preprint arXiv:1512.03385*, 2016.

Petter Holme. Core-periphery organization of complex networks. *Physical Review E*, 72(4):046111, 2005.

Bunyodbek Ibrokhimov and Justin-Youngwook Kang. Two-stage deep learning method for breast cancer detection using high-resolution mammogram images. *Applied Sciences*, 12(9):4616, 2022.

Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. Identification of influential spreaders in complex networks. *Nature physics*, 6(11):888–893, 2010.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Huan Liu, Shu Zhang, Xi Jiang, Tuo Zhang, Heng Huang, Fangfei Ge, Lin Zhao, Xiao Li, Xintao Hu, Junwei Han, et al. The cerebral cortex is bisectionally segregated into two fundamentally different functional units of gyri and sulci. *Cerebral Cortex*, 29(10):4238–4252, 2019.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Chong Ma, Lin Zhao, Yuzhong Chen, Lu Zhang, Zhenxiang Xiao, Haixing Dai, David Liu, Zihao Wu, Zhengliang Liu, Sheng Wang, et al. Eye-gaze-guided vision transformer for rectifying shortcut learning. *arXiv preprint arXiv:2205.12466*, 2022.

Inês C Moreira, Igor Amaral, Inês Domingues, António Cardoso, Maria Joao Cardoso, and Jaime S Cardoso. Inbreast: toward a full-field digital mammographic database. *Academic radiology*, 19 (2):236–248, 2012.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, 54(4):1–34, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

M Puck Rombach, Mason A Porter, James H Fowler, and Peter J Mucha. Core-periphery structure in networks. *SIAM Journal on Applied mathematics*, 74(1):167–190, 2014.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Steven Skiena. *Implementing discrete mathematics: combinatorics and graph theory with Mathematica*. Addison-Wesley Longman Publishing Co., Inc., 1991.

Xiu Su, Shan You, Jiyang Xie, Mingkai Zheng, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Vitas: Vision transformer architecture search. *arXiv preprint arXiv:2106.13700*, 2021.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12165–12174, 2022.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393 (6684):440–442, 1998.

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2964–2972, 2022.

Jiaxuan You, Jure Leskovec, Kaiming He, and Saining Xie. Graph structure of neural networks. In *International Conference on Machine Learning*, pp. 10881–10891. PMLR, 2020.

Hao Yu and Jianxin Wu. A unified pruning framework for vision transformers. *arXiv preprint arXiv:2111.15127*, 2021.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10819–10829, 2022.

Yizhen Zhang, Minkyu Choi, Kuan Han, and Zhongming Liu. Explainable semantic space by grounding language to vision with cross-modal contrastive learning. *Advances in Neural Information Processing Systems*, 34:18513–18526, 2021.

Mingjian Zhu, Yehui Tang, and Kai Han. Vision transformer pruning. *arXiv preprint arXiv:2104.08500*, 2021.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

A Appendix

A.1 Derivation of Mask Matrix M_{cp}

The size of mask matrix M_{cp} is 197 × 197 (196 patches plus 1 classification token), and it is a symmetric matrix. The derivation process of M_{cp} is as follows: suppose we use a 5 node CP graph, without the loss of the generality, the size of the corresponding adjacency matrix is 5×5. These 5 nodes will have 40, 39, 39, 39, and 39 patches, respectively. If the $(1, 2)$ element in A is 1, which means the node 1 is connecting to the node 2, the 40 patches in the node 1 are connecting to the 39 patches in the node 2, as a result, the elements in $(1 : 40, 40 : 79)$ and $(40 : 79, 1 : 40)$ of the mask matrix M_{cp} will be 1, where the $(40 : 79, 1 : 40)$ means the elements from the 40th row to 79th row, and from the 1st column to the 40th column. The elements in the last row and column of M_{cp} are 1 because the classification token is connected to all nodes, including core and periphery nodes.

A.2 The Sampling of CP graphs

For example, if the total number of nodes is 50, the numbers of core nodes are set to be [10, 20, 30, 40], and four kinds of CP graphs, including [50, 10], [50, 20], [50, 30], [50, 40] are obtained. For these 4 kinds of CP graphs, we generate 5 samples for each of them.

A.3 Experiment Settings

We trained the CP-ViT for 100 epochs with batch size 64 for INBreast and 256 for CIFAR-100 , and used cosine learning rate schedule (Loshchilov & Hutter (2016)) with an initial learning rate of 0.0001 and minimum of $1\text{e}{-7}$. All the experiments were conducted using NVIDIA Tesla V100 GPU.

A.4 Introduction of Datasets

The INbreast database Ma et al. (2022) is a mammographic database that contains images that were taken at the Breast Center in the Hospital de So Joo in Porto, Portugal. INbreast has a total of 115 cases (410 images) where 90 cases are from women with both breasts (4 images per case) and 25 cases are from mastectomy patients (2 images per case). This dataset has 3 classes: normal, benign, and malignant. The CIFAR-100 dataset (Krizhevsky et al. (2009)) consists 60000 color images in 100 classes with 600 images per class. There are 500 training images and 100 testing images for each class.